**WEB CRAWLER**

A COURSE PROJECT REPORT

By

**ANSHUL TOSHNIWAL(RA2011031010081)**

**SANSKAR BOHORA(RA2011031010111)**

**SAUNDARYA VERMA (RA2011031010091)**

**SANCHAL JAIN (RA2011031010072 )**

Under the guidance of   
**Dr. Revathi Venkataraman**

*In partial fulfilment for the Course*

of

18CSC302J - COMPUTER NETWORKS

in <NETWORKING>



**FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chenpalpattu District**

NOVEMBER 2022

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

**BONAFIDE CERTIFICATE**

Certified that this mini project report " **WEB CRAWLER** " is the bonafide work **ANSHUL TOSHNIWAL(RA2011031010081),SANSKAR BOHORA(RA2011031010111),SAUNDARYA VERMA (RA2011031010091) and SANCHAL JAIN (RA2011031010072 )**who carried out the project work under my supervision.

# SIGNATURE

<Faculty Name>

# <Designation>

**<Department>**

SRM Institute of Science and Technology

# ABSTRACT

The World Wide Web is an interlinked collection of billions of documents formatted using HTML. Ironically the very size of this collection has become an obstacle for information retrieval. The user has to shift through scores of pages to come upon the information he/she desires. Web crawlers are the heart of search engines. Web crawlers continuously keep on crawling the web and find any new web pages that have been added to the web, pages that have been removed from the web. Due to growing and dynamic nature of the web; it has become a challenge to traverse all URLs in the web documents and to handle these URLs. A focused crawler is an agent that targets a particular topic and visits and gathers only relevant web pages. In this dissertation I had worked on design and working of web crawler that can be used for copyright infringement. We will take one seed URL as input and search with a keyword, the searching result is based on keyword and it will fetch the web pages where it will find that keyword. This focused based crawler approach retrieve documents that contain particular keyword from the user's query; we are implementing this using breadth-first search. Now, when we retrieved the web pages we will apply pattern recognition over text. We will give one file as input and apply the pattern recognition algorithms. Here, pattern symbolizes text only and check how much text is available on the web page. The algorithms that I had used for pattern search are Knutt-Morri-Pratt, Boyer-Moore, finite automata algorithm.

# ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professor Dr.Annapurani Panaiyappan, Professor and Head, Department of Networking and Communications** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **<Faculty Name> , <Designation> , <Department>,** for his/herassistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD <Name> <Designation>, <Department>** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

**TABLE OF CONTENTS**

## CHAPTERS CONTENTS PAGE NO.

* + - 1. **ABSTRACT**
      2. **INTRODUCTION**
      3. **LITERATURE SURVEY**
      4. **REQUIREMENT ANALYSIS**
      5. **ARCHITECTURE & DESIGN**
      6. **IMPLEMENTATION**
      7. **EXPERIMENT RESULTS & ANALYSIS**
         1. RESULTS
         2. RESULT ANALYSIS
      8. **CONCLUSION & FUTURE ENHANCEMENT**
      9. **REFERENCES**

1. **INTRODUCTION**

**Scenario Description**

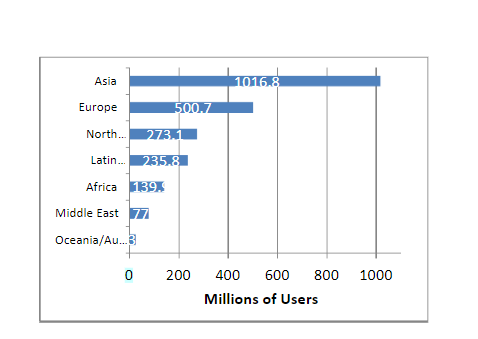
Internet is a gathering place for a large amount of information in the world, be it text, media or data in other formats that are usually displayed in a web page. The ease of data accessing is important thing for most business success in the modern world. For companies engaged in marketing, data can be used to determine current market trends, so the most appropriate marketing strategies can be found for each product. Ecommerce-based companies can also use this data for market analysis or simply price comparisons with other ecommerce competitors. Among the world's total population of 7.5 billion, 3.6 billion are internet users. It means a half of the world's population is on the internet. There are more than 1 billion websites on the World Wide Web (WWW)today. On average, Google processes more than 40,000 search queries per second, which translates to 3.5 billion search queries per day worldwide. Because of information and users on the WWW are growing at a rapid rate, it becomes a challenge for Search Engines to fill all user needs for information searching on interest topics. Typing any query in a Search Engine will generate millions of web documents. Most of these documents are irrelevant to user interests. It is very difficult for users to find relevant information from this huge collection of results. The most searched data on the Internet is news, almost everyone every day needs the latest news. Increasing news data on the internet can be a costeffective medium of information in Indonesia.

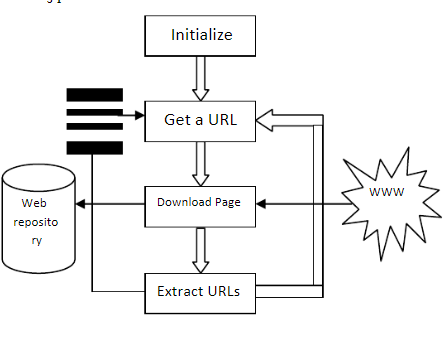
Search Engines use Web Crawlers to collect web pages from the WWW by following hyperlinks on these web pages. Web Crawler is a search engine that works by downloading a web page that passes a hyperlink on that page. Therefore crawlers are often referred to asgraphical browsing. The working principle of a crawler is starting with a list of visited URL, then the crawler will visit the URL address one by one until it's finished. Design and implementation of distributed news domain detection system on a web crawler is the first step to obtaining data in the news form which is very complex. Thus, design built can be a data source which easily accessed and used. In this study, we propose a method for gathering news online. A method called distributed focused crawler. The only result we need is news data. The main components crawlers focused are classifiers, refiners and crawlers. Classification is carried out with the aim of getting a decision on which web page matches the desired problem.

**WEB CRAWLER**

A web crawler is a program/software or programmed script that browses the World Wide Web in a systematic, automated manner. The structure of the WWW is a graphical structure, i.e., the links presented in a web page may be used to open other web pages. Internet is a directed graph where webpage as a node and hyperlink as an edge, thus the search operation may be summarized as a process of traversing directed graph. By following the linked structure of the Web, web crawler may traverse several new web pages starting from a webpage. A web crawler move from page to page by the using of graphical structure of the web pages. Such programs are also known as robots, spiders, and worms. Web crawlers are designed to retrieve Web pages and insert them to local repository. Crawlers are basically used to create a replica of all the visited pages that are later processed by a search engine 23.9 77 139.9 235.8 273.1 500.7 1016.8 0 200 400 600 800 1000 Oceania/Au… Middle East Africa Latin … North … Europe Asia Millions of Users .

Search engines job is to storing information about several webs pages, which they retrieve from WWW. These pages are retrieved by a Web crawler that is an automated Web browser that follows each link it sees.





1. **LITERATURE SURVEY**

Possibly the largest level study of Web page change was performed by Fetterly et al. [46]. They crawled 151 million pages once a week for 11 weeks, and compared the modification across pages. Like Ntoulas et. al., they found a relatively small amount of change, with 65% of all page pairs remaining exactly the same.

The study furthermore found that past change was a good judge of future change, this page length was correlated with change, and that the top level domain of a page was correlated with change. Describing the amount of change on the Web has been of significant interest to researchers .Cho and Garcia-Molina crawled around 720,000 pages once a day for a period of four months and seemed at how the pages changed. Ntoulas et. al. [50] studied page change through weekly downloaded of 154 websites collected over a year.

They found that a large number of pages did not modify according to a bags of words measure of similarity. Even for pages that did change, the changes were small. Frequency of change was not a big judge of the degree of change, but the degree of change was a good judge of the future degree of change.

More recently, Olston and Panday [51] crawled 10,000 random samples of URLs and 10,000 pages sampled from the OpenDirectory every second days for several months.

Their analysis measured both change frequency and information longevity is the average lifetime of a shingle, and found only a moderate correlation between the two.

They introduce new crawl policies that are aware to information longevity. In a study of changes examined via a proxy, Douglis et al identified an association between re visitation rates and change. Hence, the study was limited to web content visited by a restricted population, and web pages were not aggressively crawled for changes among different visits.

Researchers have also peeped at how search results modify over time .The main focus in this study was on recognizing the dynamics of the consequences change and search engines has for searchers who want to return to previously visited pages. Junghoo Cho and Hector Garcia-Molina [30] proposed the design of an effective parallel crawler. The size of the Web grows at very fast speed, it becomes essential to parallelize a crawling process, to complete downloading pages in a reasonable amount of time. Author first proposes multiple architectures for a parallel crawler and then identifies basic issues related to parallel crawling.

Based on this understanding, author then propose metrics to evaluate a parallel web crawler, and compare the proposed architectures using millions of pages collected from the Web. Rajashree Shettar, Dr. Shobha G presented a new model and architecture of the Web Crawler using multiple HTTP connections to WWW.

The multiple HTTP connection is applied using multiple threads and asynchronous downloader part so that the overall downloading process is optimum. The user gives the initial URL from the GUI provided. It begins with a URL to visit. As the crawler visits the URL, it identifies all the hyperlinks available in the web page and appends them to the list of URLs to visit, known as the crawl frontier.

URLs from the frontier is iteratively visited and it ends when it reaches more than five levels from every home pages of the websites visited and it is accomplished that it is not required to go deeper than five levels from the home page to capture most of the pages visited by the people while trying to retrieve information from the internet.

Eytan Adar et. al described algorithms, analyze, and models for characterizing the evolution of Web content. Proposed analysis gives insight into how Web content changes on a finer grain than previous study, both in terms of the time intervals studied and the detail of change analyzed.

A. K. Sharma et. al. [33] Parallelization of crawling system is necessary for downloading documents in a reasonable amount of time. The work has done reported here to focuses on providing parallelization at three levels: the document, the mapper, and the crawl worker level. The bottleneck at the document level has been removed.

The efficacy of DF (Document Fingerprint) algorithm and the efficiency of volatile information has been tested and verified. This paper specifies the major components of the crawler and their algorithmic detail. Ashutosh Dixit et. al. developed a mathematical model for crawler revisit frequency.

This model ensures that frequency of revisit will increase with the change frequency of page up to the middle threshold value after that up to the upper threshold value remains same i.e., unaffected by the change frequency of page but after the upper threshold value it starts reducing automatically and settles itself to lower threshold. Shruti Sharma et. al. present architecture for a parallel crawler which includes multiple crawling processes; called C-procs. Each C-proc performs the vital tasks that a single process crawler performs.

It downloads pages from the WWW, stores the pages locally, extracts URLs from them and follows their links. The C-proc’s executing these tasks may be spread either on the same local network or at geographically remote locations. Alex Goh Kwang Leng et. al. [36] Developed algorithm which uses the standard Breadth-First Search strategy to design and develop a Web Crawler called PyBot.

1. **REQUIREMENTS**

**CRAWLING TECHNIQUES**

There are a few crawling techniques used by Web Crawlers, mainly used are:

**A. General Purpose Crawling**

A general purpose Web Crawler collects as many pages as it can from a particular set of URL’s and their links. In this, the crawler is able to fetch a large number of pages from different locations. General purpose crawling can slow down the speed and network bandwidth because it is fetching all the pages.

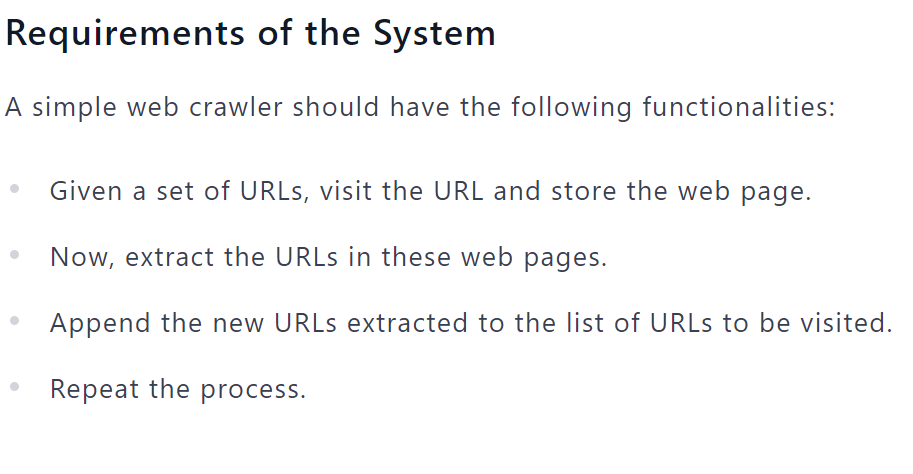
**B. Focused Crawling**

A focused crawler is designed to collect documents only on a specific topic which can reduce the amount of network traffic and downloads. The purpose of the focused crawler is to selectively look for pages that are appropriate to a pre-defined set of matters. It crawl only the relevant regions of the web and leads to significant savings in hardware and network resources. Initialize Get a URL Download Page Extract URLs WWW Web repository.

**C. Distributed Crawling**

In distributed crawling, multiple processes are used to crawl and download pages from the Web

**Requirement Analysis**



**Steps to Build A Web Crawler Using Python:**

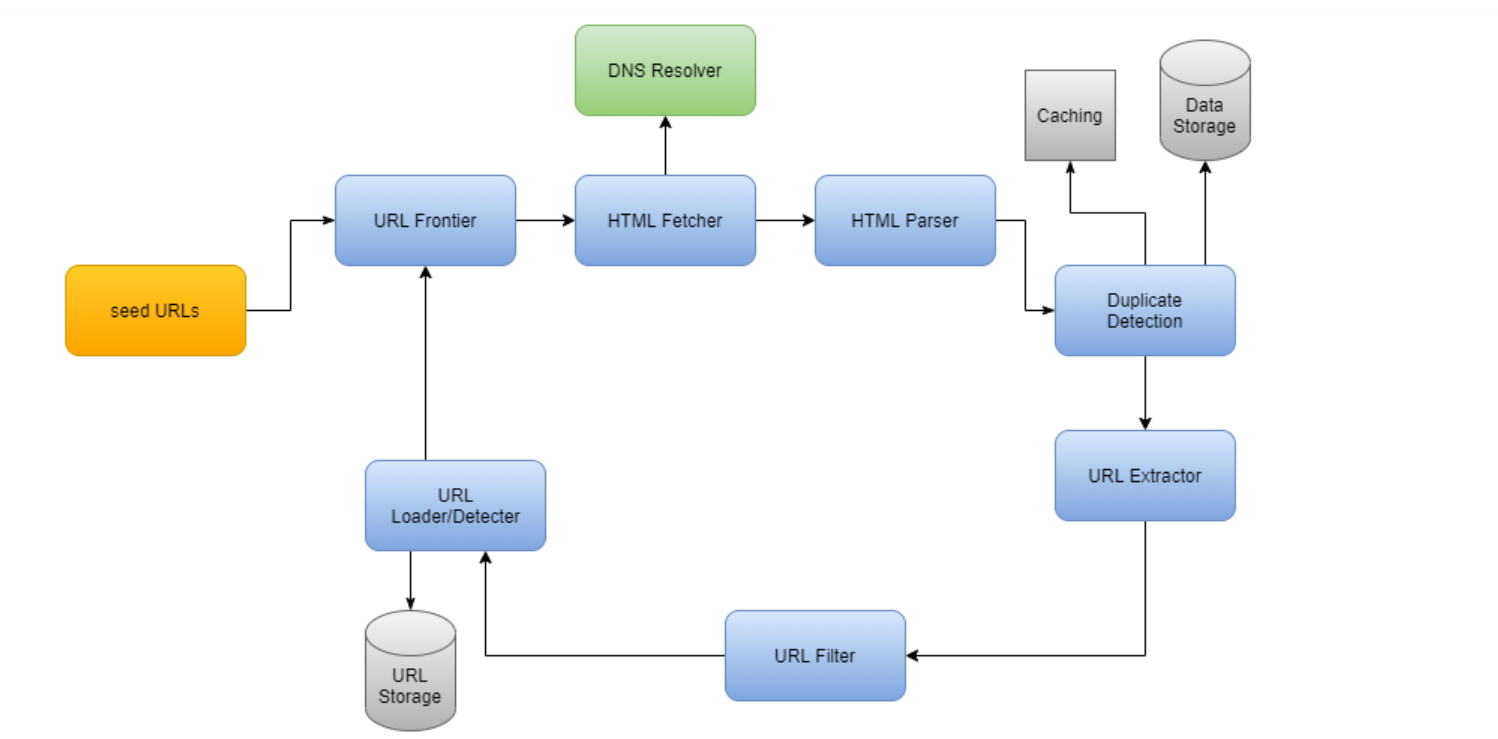
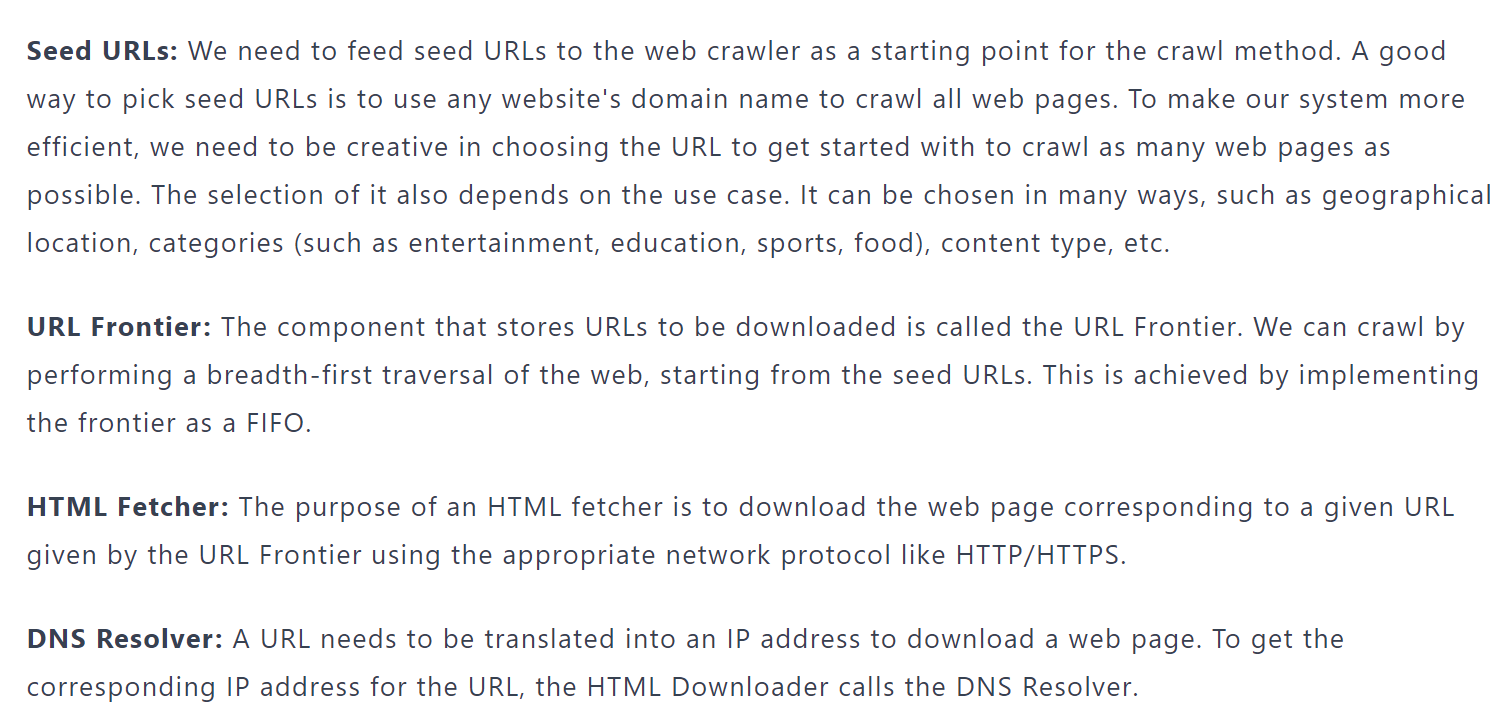
Step 1: Send an HTTP request to the URL of the webpage. It responds to your request by returning the content of web pages.

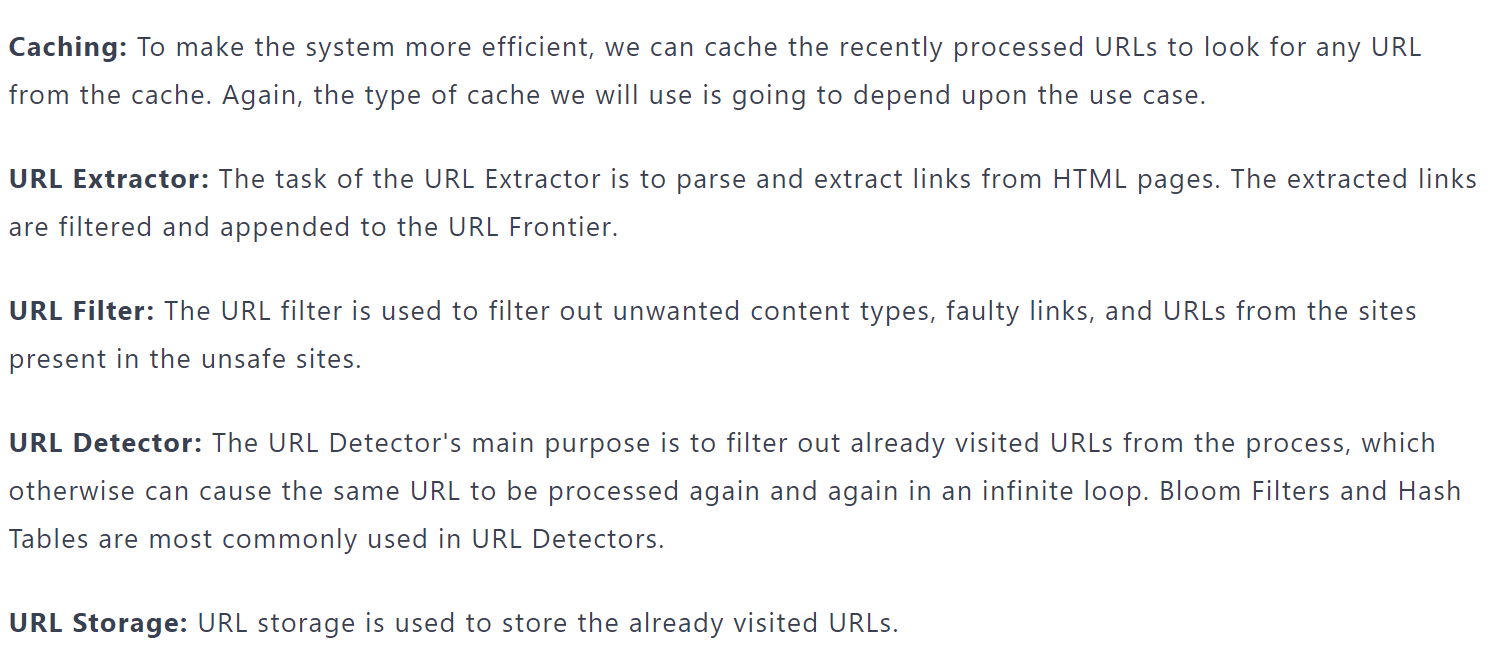
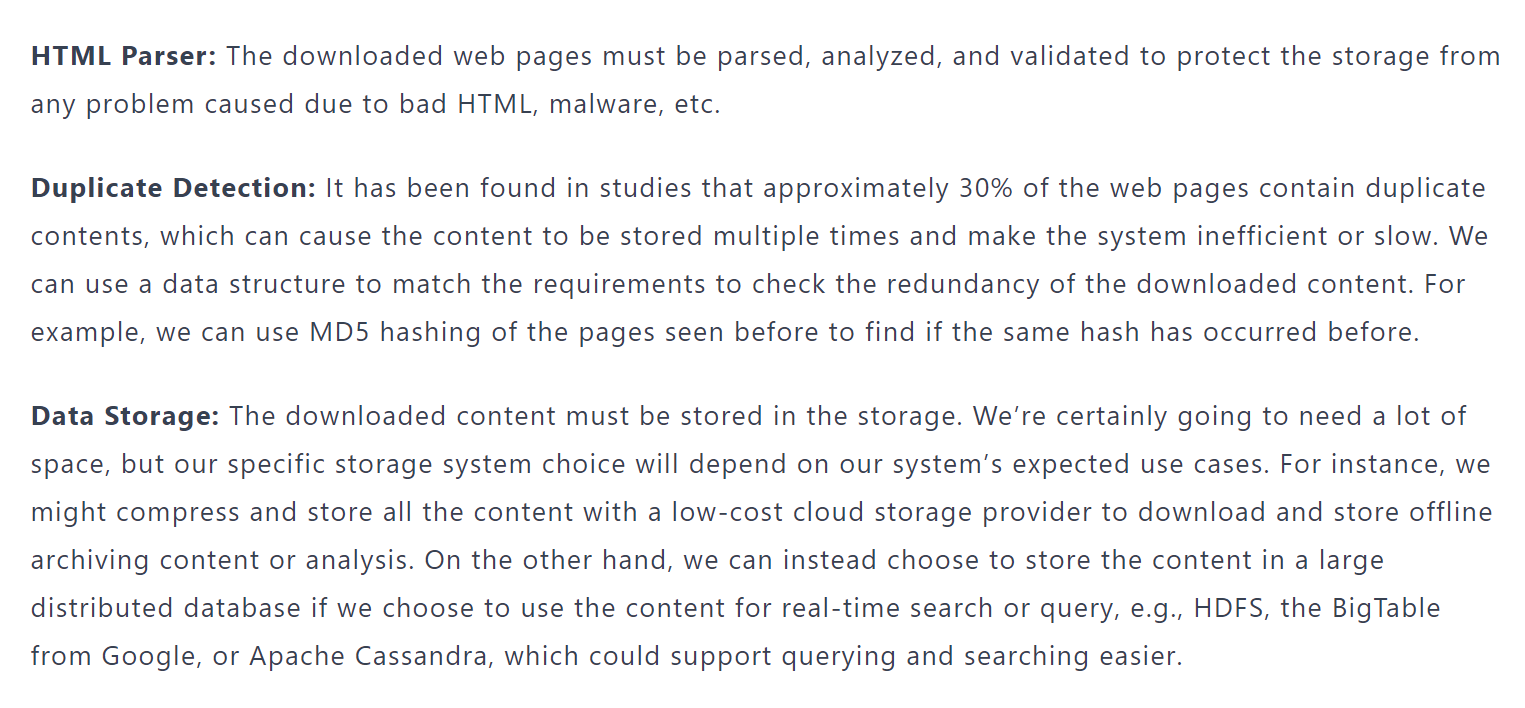
Step 2: Parse the webpage. A parser will create a tree structure of the HTML as the webpages are intertwined and nested together. A tree structure will help the bot follow the paths that we created and navigate through to get the information.

Step 3: Using the Python library to search the parse tree.

Among the computer languages for a web crawler, Python is easy-to-implement compared to PHP and Java. It still has a steep learning curve that prevents many non-tech professionals from using it. Even though it is an economic solution to write your own, it's still not sustainable with regard to the extended learning cycle within a limited time frame.

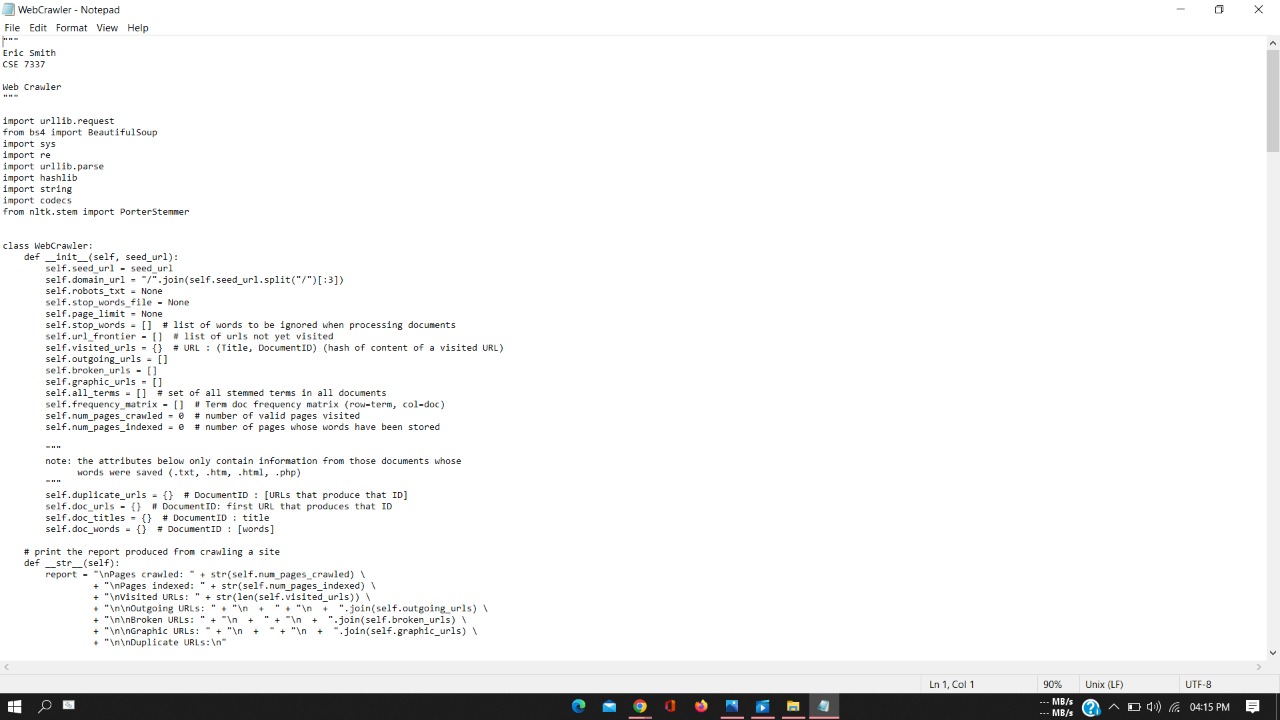
1. **ARCHITECTURE AND DESIGN**

****

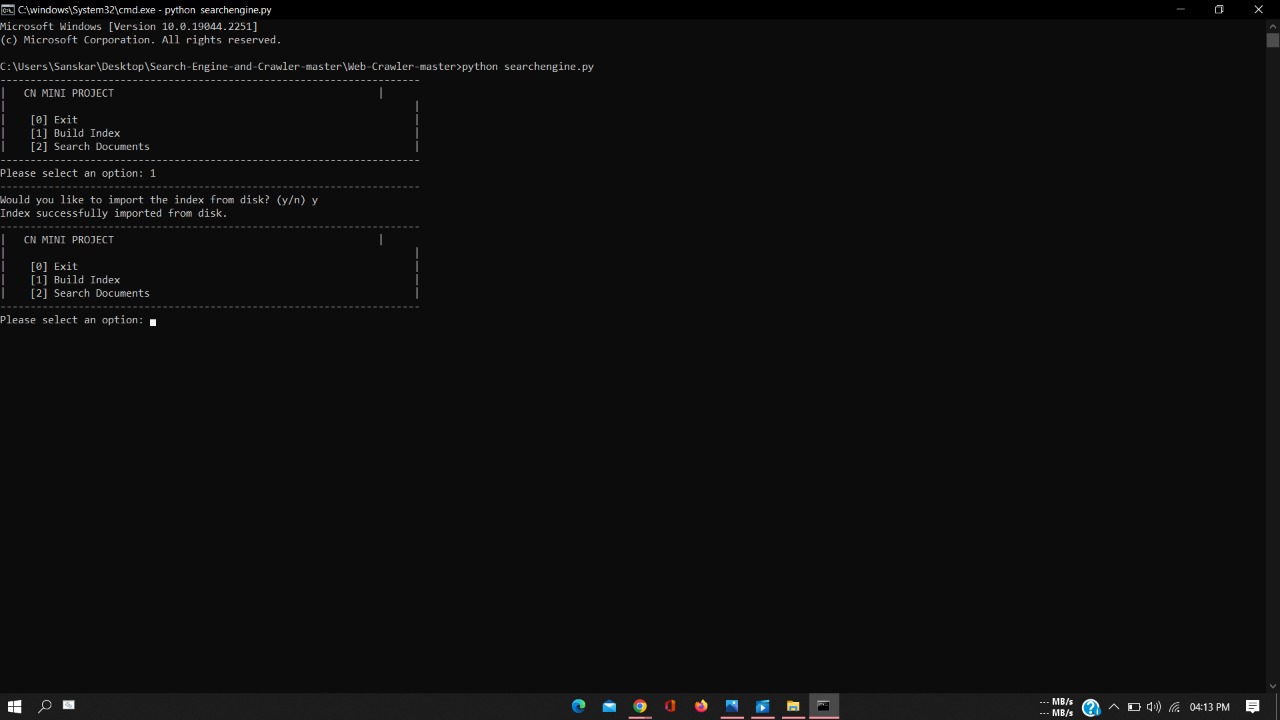


1. **IMPLEMENTATION**

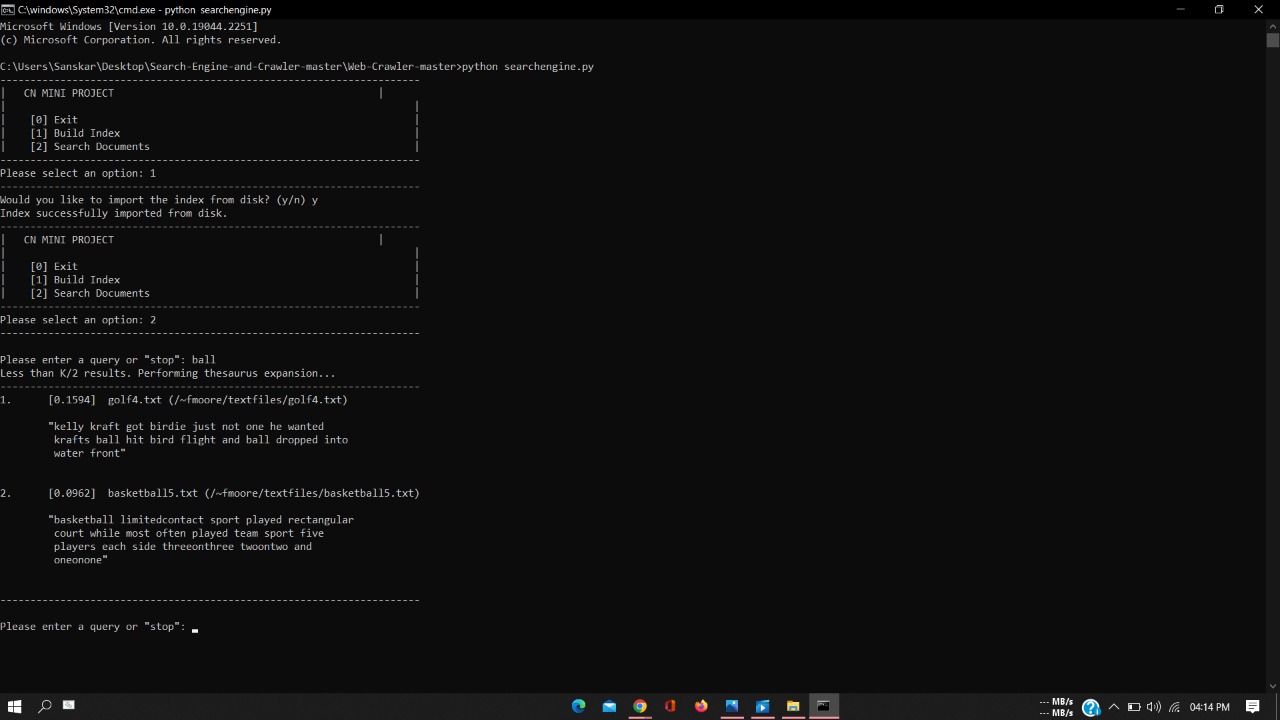
To develop and implement this system used python programming language and its default libraries. Data extraction, read-write and stored functions are fetched and imply by python raw code. Any web site can adopt this system to use web-crawling technology to collect data from World Wide Web (www).



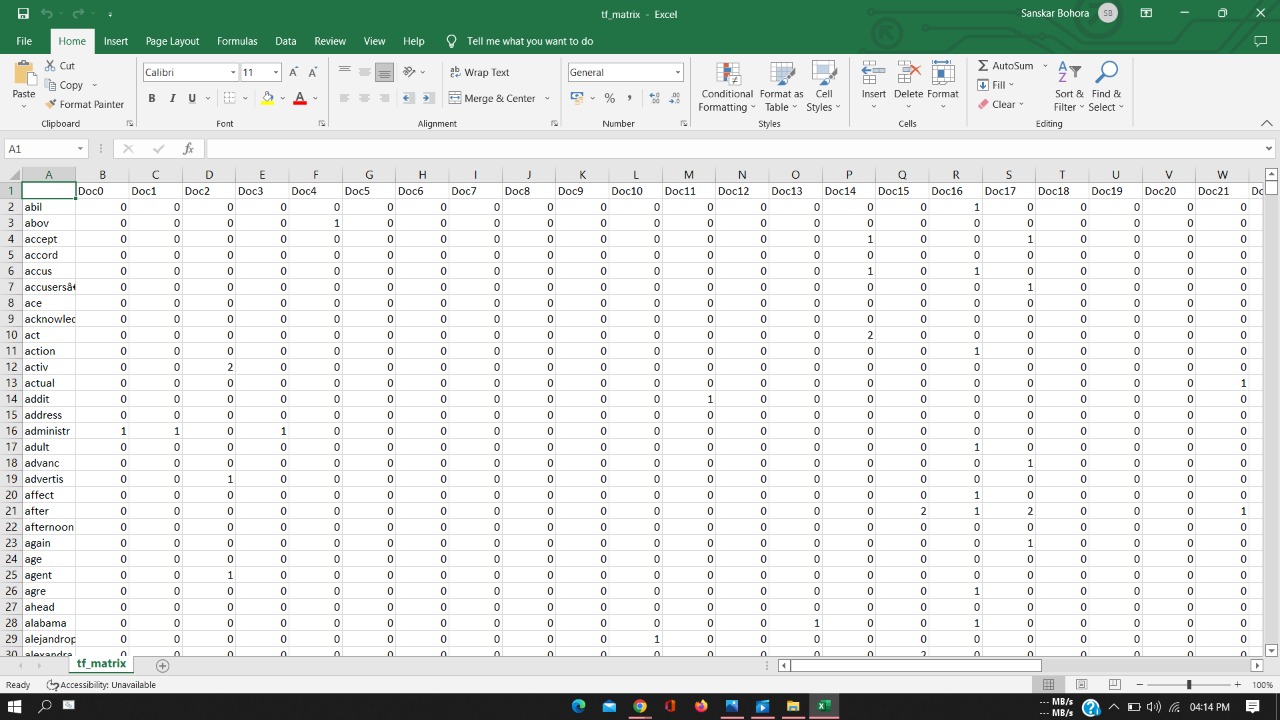
In Figure 3, at first, I have imported the libraries' requests, BeautifulSoup, and request. Then a function named crawler\_web() with the parameter max\_pages is declared. In the function first a variable is declared and initialized with the value 1. A while loop is implemented with the condition that the loop is true as long as the variable ‗page‘ is less than ‗max\_page‘. In the loop an URL of a webpage is set from where information is to be extracted. We create an object ‗source\_code‘ for which we get all the information where we save the returned values of the HTTP request. The values are converted to plain text and save it in variable ‗plain\_text‘. To pull out the information (HTML file) using ‗BeautifulSoup‘ we create an object. Now a ‗for loop‘ is used traverse through the information (HTML file) that is pulled out. If in the file there is a class ‗btn bth-primary data-link‘ the link next to it will be extracted, then concat with the text ‗http://www.data.gov.bd‘ and saved in the variable href. The variable title while saving the string of the link. The title and the link will be printed out in the output console. After the for loop ends, the variable ‗page‘ will be incremented with 1 and the while loop will continue. The function ‗crawler\_web()‘ is called with argument ‗1‘.



In Figure 4, we can see, in the HTML file, the link did not contain any text so there is no title and the variable contained nothing and ‗None‘ is printed out. In the next line, we see an HTTP link that the web crawler extracted from the webpage. All the links on the page are extracted.

****

In Figure 5, the library ‗request‘ is imported from ‗urllib‘. Then a link extracted by the web crawler in saved in the variable ‗notice\_url‘. A the function ‗download\_csv()‘ with parameter ‗csv\_url‘. The function is called with ‗download\_csv()‘ with the argument ‗notice\_url‘. The function receives the link as its parameter. An object ‗response‘ is created to extract the information from the link. The information is read and stored in variable ‗csv‘. The information is converted to string format and stored in variable ‗csv\_str‘. The strings are then split into lines and saved in ‗lines‘. Now an object named ‗dest\_url‘ is created to create a CSV file named ‘dataset.csv‘. With the object fx, the ‗dest\_url‘ file is opened and written on. Using a ‗for‘ loop each line from ‗line‘ variable is written on the CSV file. Then the file is closed and stored.



1. **RESULTS AND DISCUSSION**

Web crawlers have a basic technique of simulating the browser to design HTTP requests, and the crawler will send requests to a Web server via HTTP. The crawler will analyze and store. Web page, and finish the crawler system crawling job after getting a response from the server. Basically decomposing website pages is a process of reducing the distractions that exist on website pages. On the Internet, all categories of web page information are stored in an HTML framework. In fact, web page denoising is simply the extraction of text from web content. When a theme crawler extracts content on a web page, it needs to decipher the HTML structure of the page in order to effectively extract information from the page. Common methods include parsing the HTML structure of Beautiful Soup and extracting the text data using a regular expression.

1. **CONCLUSION AND FUTURE ENHANCEMENT**

The Internet and Intranets have brings a lots of information. People usually have the option to search engines to find necessary information. Web Crawler is thus vital information retrieval which traverses the Web and downloads web documents that suit the user's need. Web crawlers are designed to retrieve Web pages and insert them to local repository. Crawlers are basically used to create a replica of all the visited pages which are later processed by a search engine that will index the downloaded pages that help in quick searches. The major objective of the review paper is to throw some light on the web crawling previous work. This article also discussed the various researches related to web crawler.

**REFERENCES**

[1] Berners-Lee, Tim, “The World Wide Web: Past, Present

and Future”, MIT USA, Aug 1996, available at:

http://www.w3.org/People/Berners-Lee/1996/ppf.html.

[2] Berners-Lee, Tim, and Cailliau, CN, R., “Worldwide

Web: Proposal for a Hypertext Project” CERN October

1990, available at: http://www.w3.org/Proposal.html.

[3] “Internet World Stats. Worldwide internet users”,

available at: http://www.internetworldstats.com

(accessed on May 5, 2012).

[4] Maurice de Kunder, “Size of the World Wide Web”,

Available at: http://www.worldwidewebsize.com

(accessed on May 5, 2012).

[5] P. J. Deutsch. Original Archie Announcement, 1990.

URL

http://groups.google.com/group/comp.archives/msg/a773

43f9175b24c3?output=gplain.

[6] A. Emtage and P. Deutsch. Archie: An Electronic

Directory Service for the Internet. In roceedings of the

Winter 1992 USENIX Conference, pp. 93–110, San

Francisco, California, USA, 1991